Homework 2: Sentiment

Course: CS 221 Spring 2019
Name: Bryan Yaggi

Problem 1: Building Intuition

Rotten Tomatoes has classified these reviews as "positive" and "negative", respectively, as indicated by the intact tomato on the left and the splattered tomato on the right. In this assignment, you will create a simple text classification system that can perform this task automatically. We'll warm up with the following set of four mini-reviews, each either labeled positive (+1) or negative (−1):

1. (−1) pretty bad
2. (+1) good plot
3. (−1) not good
4. (+1) pretty scenery

Each review x is mapped onto a feature vector $\phi(x)$, which maps each word to the number of occurrences of that word in the review. For example, the first review maps to the (sparse) feature vector $\phi(x) = \{pretty : 1, bad : 1\}$. Recall the definition of the hinge loss:

$$Loss_{hinge}(x, y, \mathbf{w}) = max\{0, 1 - \mathbf{w} \cdot \phi(x)y\},$$

where y is the correct label.

(a) Suppose we run stochastic gradient descent, updating the weights according to

$$w \leftarrow w - \eta \nabla_w Loss_{hinge}(x, y, \mathbf{w}),$$

once for each of the four examples in the order given above. After the classifier is trained on the given four data points, what are the weights of the six words ("pretty", "good", "bad", "plot", "not", "scenery") that appear in the above reviews? Use $\eta = .5$ as the step size and initialize $w = [0, \ldots, 0]$. Assume that $\nabla_w Loss_{hinge}(x, y, \mathbf{w}) = 0$ when the margin is exactly 1. “”

Let the feature and weights vector be the number of times “pretty”, “bad”, “bad”, “plot”, “not”, and “scenery” occur in x, respectively.

$$x_1 = \text{“}pretty\ bad\text{”}, \phi(x_1) = \{\text{”}pretty\text{”} = 1, \text{”}bad\text{”} = 1\}, y_1 = -1$$
$$x_2 = \text{“}good\ plot\text{”}, \phi(x_2) = \{\text{”}good\text{”} = 1, \text{”}plot\text{”} = 1\}, y_2 = +1$$
$$x_3 = \text{“}not\ good\text{”}, \phi(x_3) = \{\text{”}not\text{”} = 1, \text{”}good\text{”} = 1\}, y_3 = -1$$
$$x_4 = \text{“}pretty\ scenery\text{”}, \phi(x_4) = \{\text{”}pretty\text{”} = 1, \text{”}scenery\text{”} = 1\}, y_4 = +1$$

$$\frac{\partial Loss_{hinge}}{\partial w} = \begin{cases} -\phi(x)y, & w \cdot \phi(x)y < 1 \\ 0, & w \cdot \phi(x)y \geq 1 \end{cases}$$

$$w_{after\ step\ 1} = \begin{bmatrix}0\\0\\0\\0\\0\\0\end{bmatrix} - .5\begin{bmatrix}(-1)(-1)\\0\\(-1)(-1)\\0\\0\\0\end{bmatrix} = \begin{bmatrix}-.5\\0\\-.5\\0\\0\\0\end{bmatrix}$$

$$w_{after\ step\ 2} = \begin{bmatrix}-.5\\0\\-.5\\0\\0\\0\end{bmatrix} - .5\begin{bmatrix}0\\(-1)(1)\\0\\(-1)(1)\\0\\0\end{bmatrix} = \begin{bmatrix}-.5\\.5\\-.5\\.5\\0\\0\end{bmatrix}$$

$$w_{after\ step\ 3} = \begin{bmatrix}-.5\\.5\\-.5\\.5\\0\\0\end{bmatrix} - .5\begin{bmatrix}0\\(-1)(-1)\\0\\0\\(-1)(-1)\\0\end{bmatrix} = \begin{bmatrix}-.5\\0\\-.5\\.5\\-.5\\0\end{bmatrix}$$

$$w_{after\ step\ 4} = \begin{bmatrix}-.5\\0\\-.5\\.5\\-.5\\0\end{bmatrix} - .5\begin{bmatrix}(-1)(1)\\0\\0\\0\\0\\(-1)(1)\end{bmatrix} = \begin{bmatrix}0\\0\\-.5\\.5\\-.5\\.5\end{bmatrix}$$

(b) Create a small labeled dataset of four mini-reviews using the words "not", "good", and "bad", where the labels make intuitive sense. Each review should contain one or two words, and no repeated words. Prove that no linear classifier using word features can get zero error on your dataset. Remember that this is a question about classifiers, not optimization algorithms; your proof should be true for any linear classifier, regardless of how the weights are learned. After providing such a dataset, propose a single additional feature that we could augment the feature vector with that would fix this problem. (Hint: think about the linear effect that each feature has on the classification score.)

Let ϕ_1, ϕ_2, ϕ_3 be the number of times "bad", "good", and "not" occur in x, respectively.

$$x_1 = \text{“}bad\text{”}, \phi_1 = \{\text{”}bad\text{”} = 1\}, y_1 = -1$$
$$x_2 = \text{“}good\text{”}, \phi_2 = \{\text{”}good\text{”} = 1\}, y_2 = +1$$
$$x_3 = \text{“}not\ bad\text{”}, \phi_3 = \{\text{”}not\text{”} = 1, \text{”}bad\text{”} = 1\}, y_3 = +1$$
$$x_4 = \text{“}not\ good\text{”}, \phi_4 = \{\text{”}not\text{”} = 1, \text{”}good\text{”} = 1\}, y_4 = -1$$

$$\begin{bmatrix}1&0&0\\0&1&0\\1&0&1\\0&1&1\end{bmatrix} \cdot \begin{bmatrix}w_1\\w_2\\w_3\end{bmatrix} - \begin{bmatrix}-1\\1\\1\\-1\end{bmatrix} \neq 0$$

Add a feature for word count. The feature will be 0 if x has one word, and 1 if x has two words.

$x_1 =$ "bad", $\phi_1 = \{$"bad" $= 1\}, y_1 = -1$
$x_2 =$ "$good$", $\phi_2 = \{$"$good$" $= 1\}, y_2 = +1$
$x_3 =$ "$not\ bad$", $\phi_3 = \{$"not" $= 1,$"bad" $= 1, count = 1\}, y_3 = +1$
$x_4 =$ "$not\ good$", $\phi_4 = \{$"not" $= 1,$"$good$" $= 1, count = 1\}, y_4 = -1$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = 0$$

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} or \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$